VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“Jnana Sangama”, Machhe, Belagavi, Karnataka-590018

Lab Experiment Record

Project Management with Git [BCSL358C]

Submitted in partial fulfillment towards AEC of 3rd semester of

Bachelor of Engineering
in
Computer Science and Engineering
(Artificial Intelligence & Machine Learning)

Submitted by

YOGITHA L
4GW24CI063

DEPARTMENT OF CSE (Artificial Intelligence & Machine Learning)

GSSS INSTITUTE OF ENGINEERING & TECHNOLOGY FOR WOMEN

(Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi & Govt. of Karnataka)

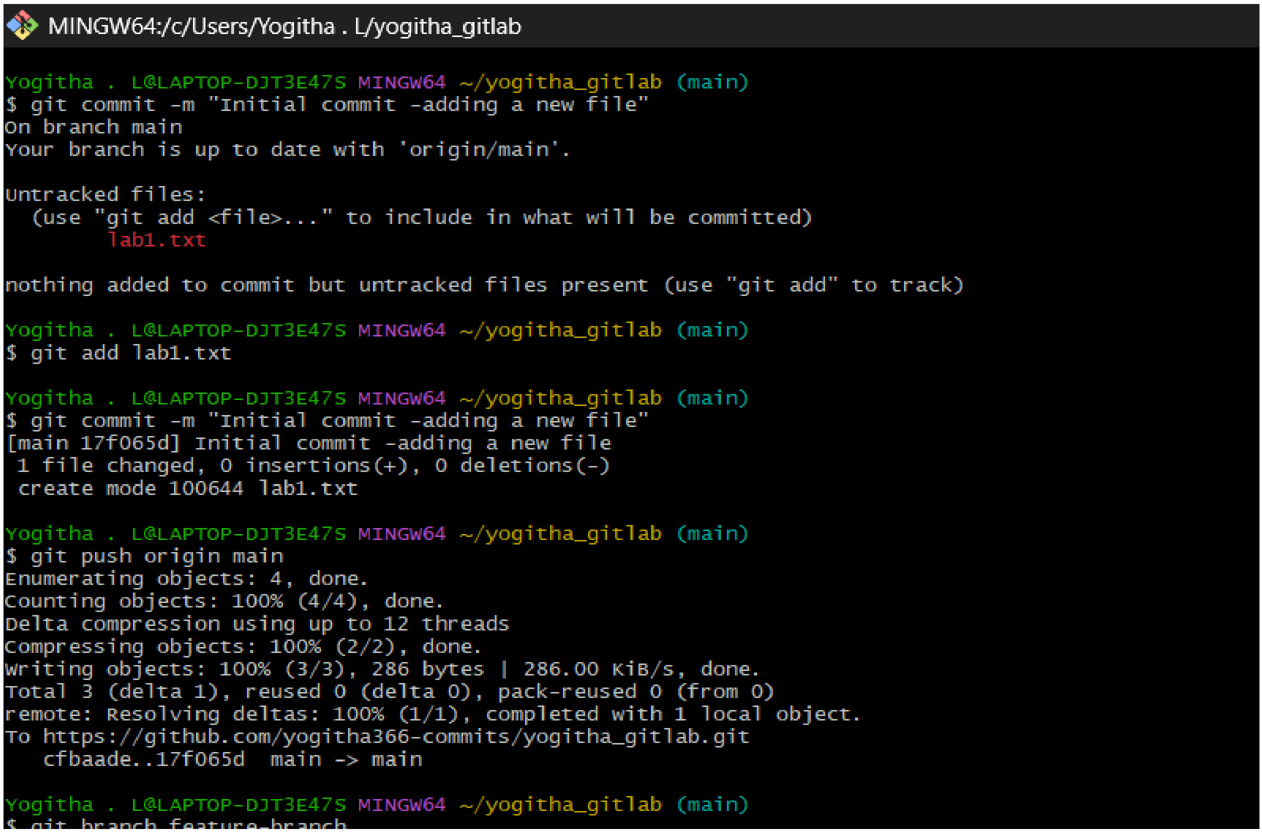
EXPERIMENT 1: SETTING UP AND BASIC COMMANDS

PROCEDURE

1. Open Git Bash and move to the project folder created on the Desktop using the cd command.
2. Initialize the folder as a Git repository using the git init command.
3. Create a new text file using Notepad to store the experiment content.
4. Add the created file to the staging area using the git add command.
5. Check the status of the repository using the git status command.
6. Commit the staged file to the local repository using the git commit command.
7. Link the local repository to the remote GitHub repository using the git remote add origin command.
8. Rename the default branch to main using the git branch M main command.
9. Synchronize the local repository with the remote repository using the git pull --rebase command.
10. Push the committed files from the local repository to the GitHub repository using the git push command.

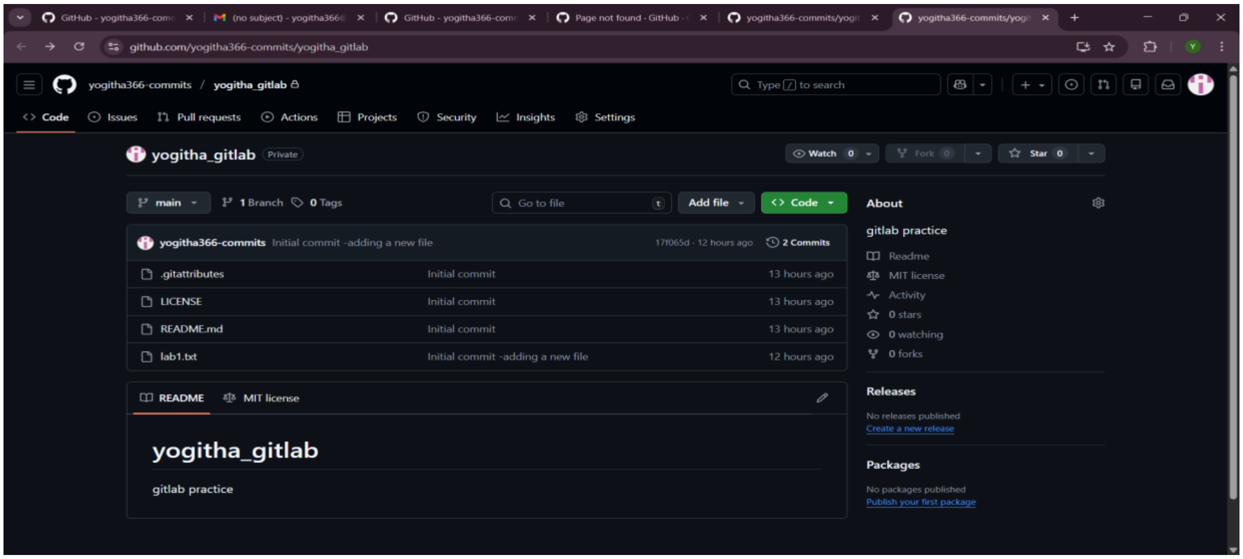
GIT REPO URL :

https://github.com/yogitha366-commits/yogitha_gitlab.git



MINGW64:/c/Users/Yogitha . L/yogitha_gitlab

```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (main)
$ git commit -m "Initial commit -adding a new file"
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        lab1.txt

nothing added to commit but untracked files present (use "git add" to track)

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (main)
$ git add lab1.txt

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (main)
$ git commit -m "Initial commit -adding a new file"
[main 17f065d] Initial commit -adding a new file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 lab1.txt

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 286 bytes | 286.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/yogitha366-commits/yogitha_gitlab.git
   cfbaade..17f065d  main -> main

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (main)
$ git branch feature-branch
```



yogitha366-commits Initial commit -adding a new file 17f065d · 12 hours ago 2 Commits

.gitattributes	Initial commit	13 hours ago
LICENSE	Initial commit	13 hours ago
README.md	Initial commit	13 hours ago
lab1.txt	Initial commit -adding a new file	12 hours ago

About: gitlab practice

yogitha_gitlab

gitlab practice

COMMANDS USED:

```
 git init notepad lab1.txt
 git add lab1.txt
 git status
git commit -m "Experiment 1: Initial repository setup and first commit"
git remote add origin https://github.com/chinmayimohan2006-web/github.git
git branch -M main git pull --rebase origin main
git push -u origin main
```

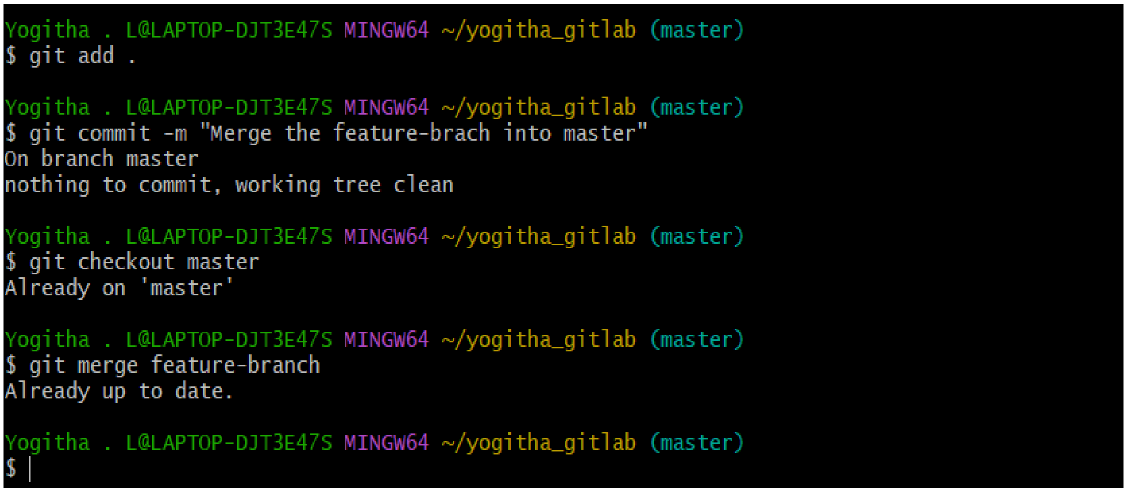
EXPERIMENT 2: CREATING AND MERGING BRANCHES

PROCEDURE

1. Open Git Bash and move to the project directory.
2. Switch to the main branch using the git checkout command.
3. Create a new branch named feature-branch and switch to it.
4. Modify an existing file using Notepad.
5. Add the modified file to the staging area.
6. Commit the changes in the feature branch.
7. Switch back to the main branch.
8. Merge the feature-branch into the main branch.

COMMANDS USED:

```
git checkout main git checkout -b feature-branch notepad
lab1.txt git add lab1.txt git commit -m "Experiment 2:
Changes in feature branch" git checkout main git merge
feature-branch
```



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git add .

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git commit -m "Merge the feature-brach into master"
On branch master
nothing to commit, working tree clean

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git checkout master
Already on 'master'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git merge feature-branch
Already up to date.

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ |
```

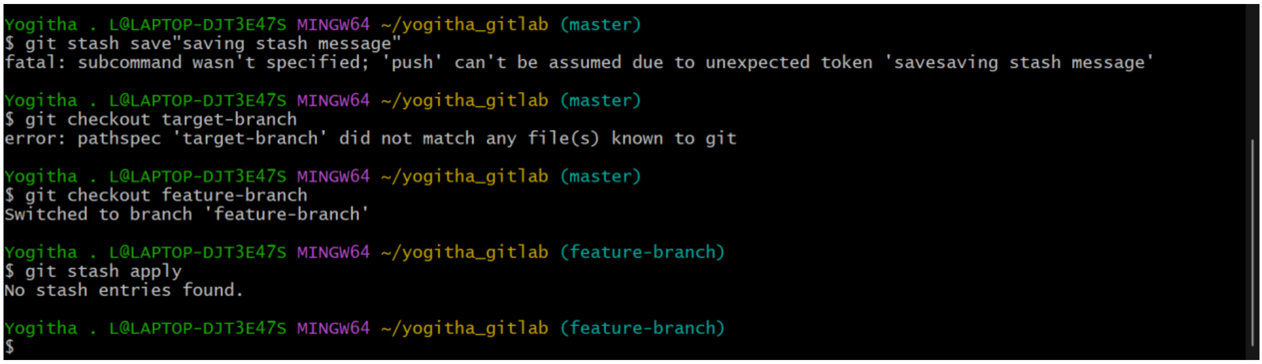
EXPERIMENT 3: CREATING AND MERGING BRANCHES

PROCEDURE:

1. Open Git Bash and move to the project directory.
2. Modify an existing file using Notepad.
3. Check the repository status using the git status command.
4. Save the uncommitted changes using the git stash command.
5. Verify that the working directory is clean.
6. Switch to another branch.
7. Apply the stashed changes using the git stash apply command.
8. Check the repository status again.

COMMANDS USED:

notepad lab1.txt , git status , git stash , git checkout feature1 , git stash apply , git status



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git stash save"saving stash message"
fatal: subcommand wasn't specified; 'push' can't be assumed due to unexpected token 'savesaving stash message'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git checkout target-branch
error: pathspec 'target-branch' did not match any file(s) known to git

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git checkout feature-branch
Switched to branch 'feature-branch'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git stash apply
No stash entries found.

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$
```

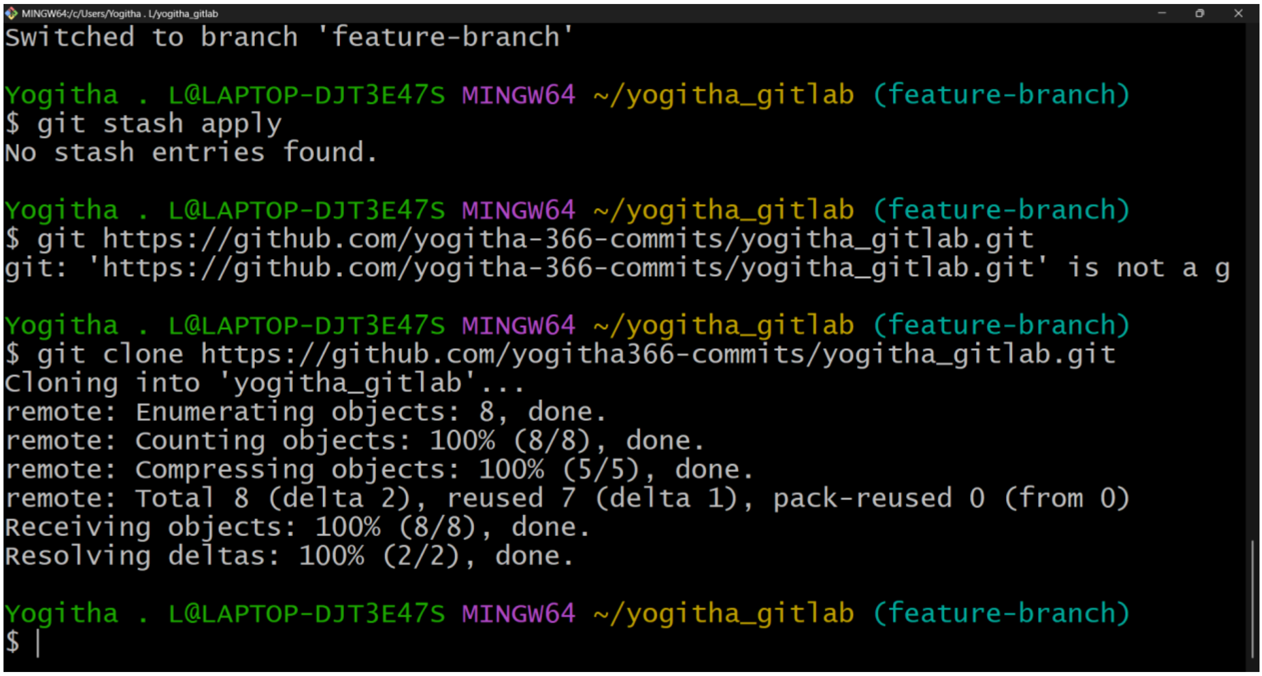
EXPERIMENT 4: COLLABORATIONS AND REPORT REPOSITORIES

PROCEDURE:

1. The terminal was navigated to the Desktop using the cd command.
2. The remote GitHub repository was copied to the local system using the git clone command.
3. The cloned repository folder was accessed using the cd command.
4. The files present in the repository were displayed using the ls command.
5. The current status of the repository was checked using the git status command.

COMMANDS USED:

```
 cd ~/OneDrive/Desktop git
clone <repository-URL>
cd <repository-folder>
ls git
status
```



```
Switched to branch 'feature-branch'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git stash apply
No stash entries found.

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git https://github.com/yogitha-366-commits/yogitha_gitlab.git
git: 'https://github.com/yogitha-366-commits/yogitha_gitlab.git' is not a g

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git clone https://github.com/yogitha366-commits/yogitha_gitlab.git
Cloning into 'yogitha_gitlab'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 2), reused 7 (delta 1), pack-reused 0 (from 0)
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (2/2), done.

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ |
```

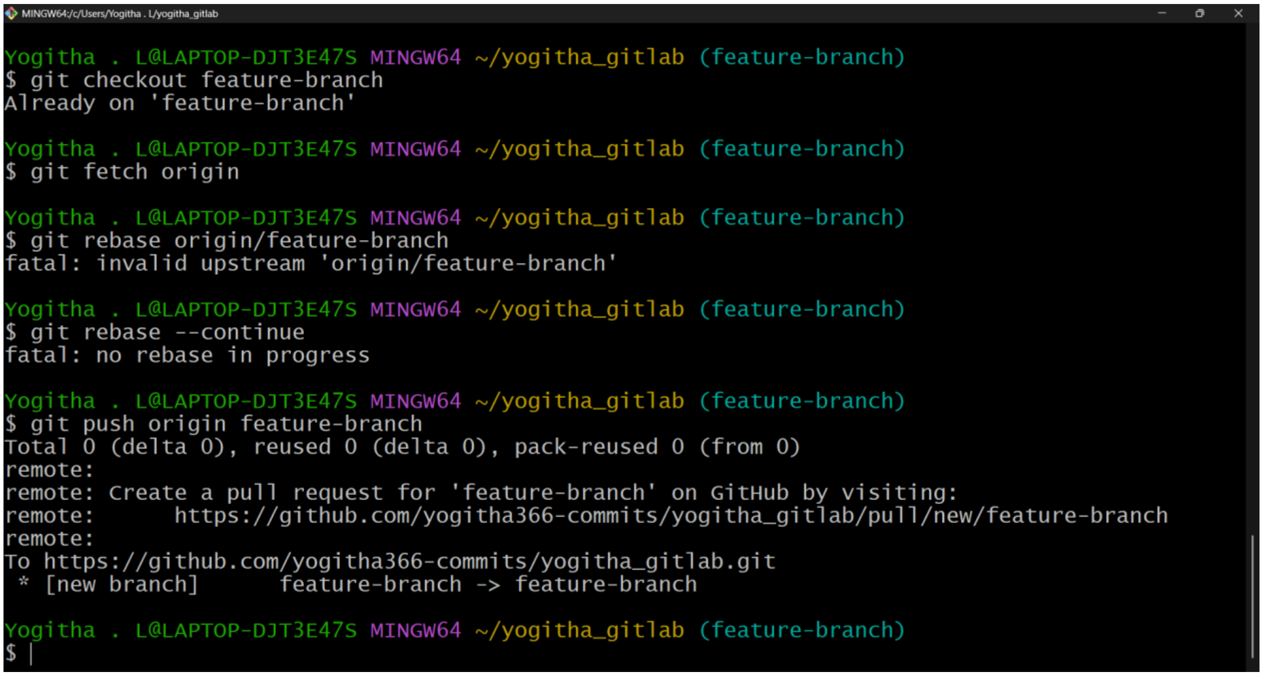
EXPERIMENT 5: COLLABORATIONS AND REPORT REPOSITORIES

PROCEDURE:

1. The terminal was navigated to the local GitHub repository directory.
2. The current branch and working tree status were verified using the git status command.
3. The latest changes from the remote repository were fetched using the git fetch origin command.
4. The local branch was synchronized with the remote main branch using the git rebase origin/main command.
5. The repository status was checked again to ensure that the working tree was clean and up to date.

COMMANDS USED:

cd ~/OneDrive/Desktop/github , git status , git status , git fetch origin, git rebase origin/main



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git checkout feature-branch
Already on 'feature-branch'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git fetch origin

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git rebase origin/feature-branch
fatal: invalid upstream 'origin/feature-branch'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git rebase --continue
fatal: no rebase in progress

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ git push origin feature-branch
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-branch' on GitHub by visiting:
remote:      https://github.com/yogitha366-commits/yogitha_gitlab/pull/new/feature-branch
remote:
To https://github.com/yogitha366-commits/yogitha_gitlab.git
 * [new branch]      feature-branch -> feature-branch

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branch)
$ |
```

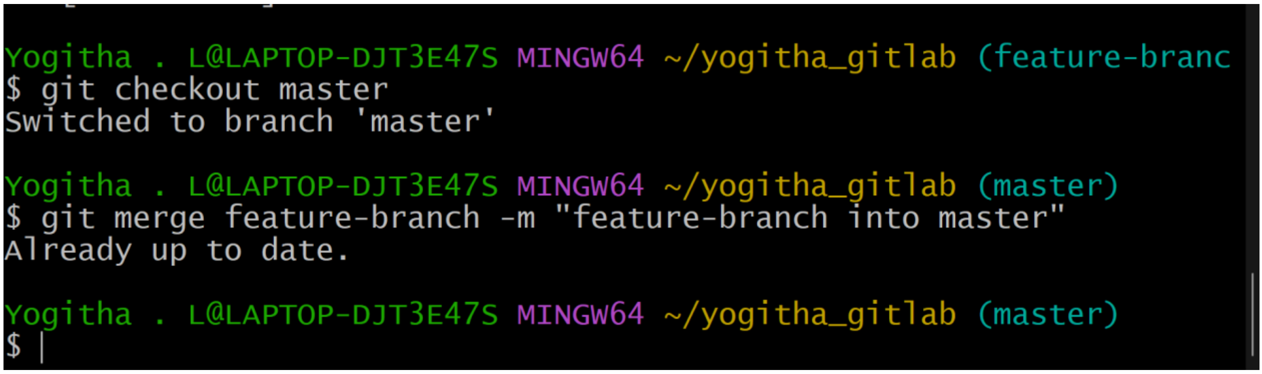
EXPERIMENT 6: COLLABORATION AND REMOTE REPOSITORIES

PROCEDURE:

1. The list of available branches was displayed using the git branch command.
2. The current working branch was switched to the main branch using git checkout main.
3. The changes from the feature branch were merged into the main branch using the git merge feature1 command.
4. The status of the repository was verified using the git status command to ensure successful merge.

COMMANDS USED:

git branch, git checkout main, git , git merge feature1, git status



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (feature-branc
$ git checkout master
Switched to branch 'master'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git merge feature-branch -m "feature-branch into master"
Already up to date.

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ |
```

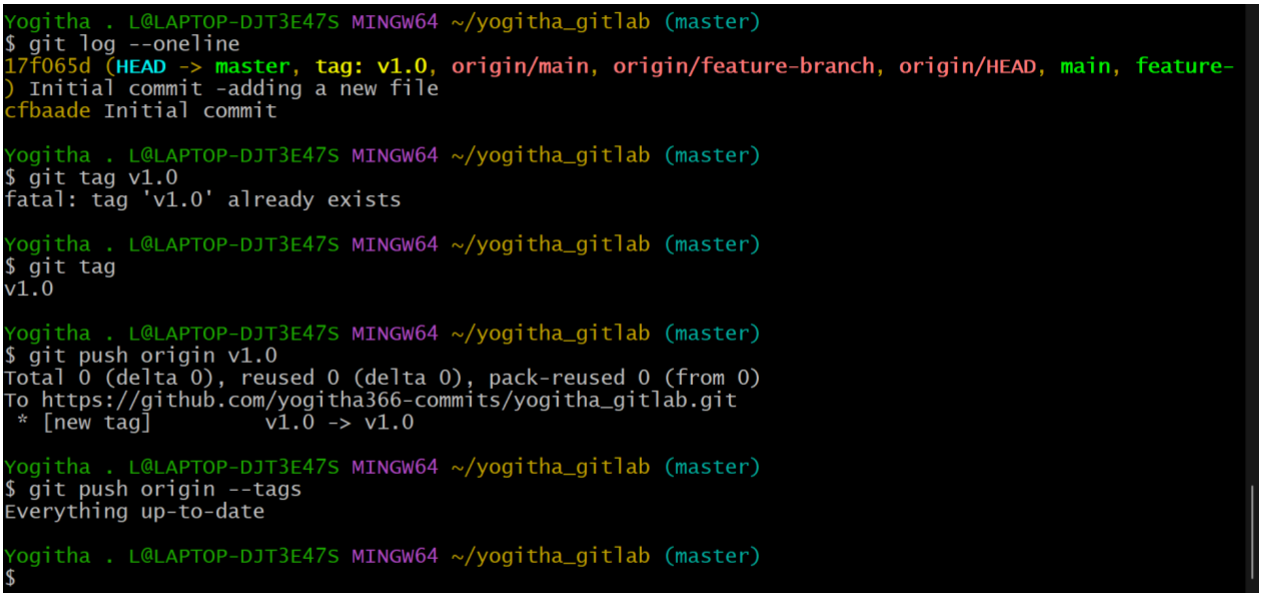
EXPERIMENT 7: GIT TAGS AND RELEASES

PROCEDURE:

1. The working branch was verified and switched to the main branch using the git checkout main command.
2. The current repository status was checked using git status.
3. A tag named v1.0 was created using the git tag command to mark an important version of the project.
4. The list of available tags was displayed using the git tag command.
5. The commit history was viewed using git log --oneline to confirm that the tag was attached to the correct commit.

COMMANDS USED:

git checkout main , git status, git tag v1.0, git tag , git log –oneline



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git log --oneline
17f065d (HEAD -> master, tag: v1.0, origin/main, origin/feature-branch, origin/HEAD, main, feature-
) Initial commit -adding a new file
cfbaade Initial commit

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git tag v1.0
fatal: tag 'v1.0' already exists

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git tag
v1.0

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git push origin v1.0
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/yogitha366-commits/yogitha_gitlab.git
 * [new tag]         v1.0 -> v1.0

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git push origin --tags
Everything up-to-date

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$
```

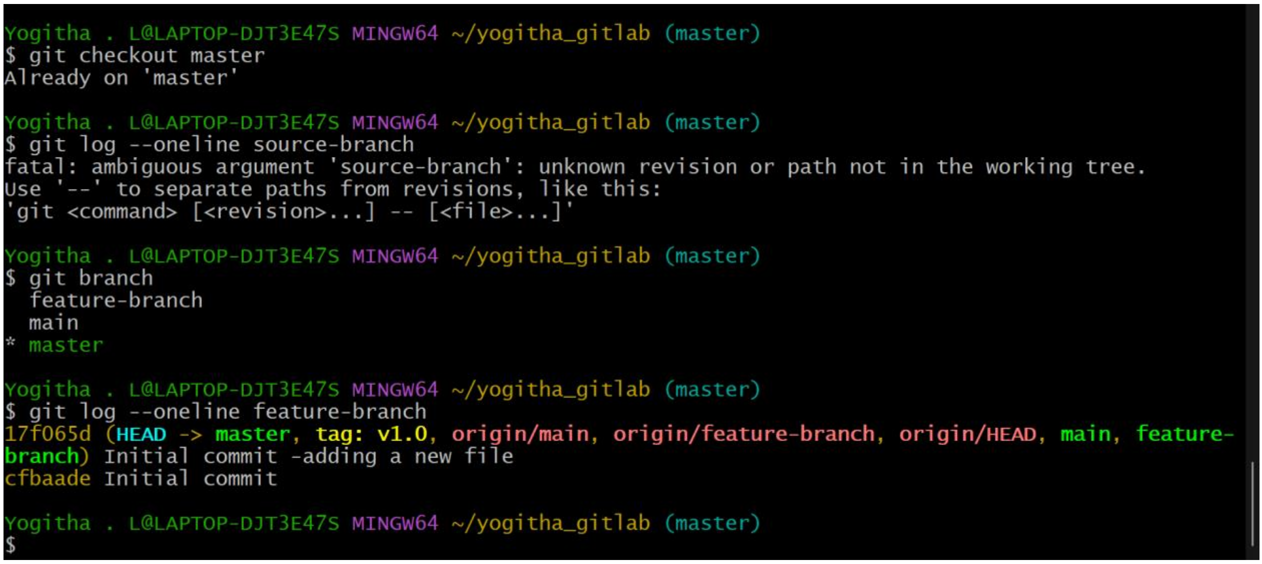
EXPERIMENT 8: ADVANCED GIT OPERATIONS

PROCEDURE:

1. The commit history was viewed using the git log --oneline command.
2. A new branch named feature1 was created using git checkout -b feature1.
3. The file lab1.txt was modified using Notepad.
4. The modified file was staged using the git add command.
5. The changes were committed using the git commit command.
6. The working branch was switched back to main using git checkout main.
7. The commit from the feature branch was applied to the main branch using the git cherrypick command.
8. The commit history was verified to confirm successful cherry-pick.

COMMANDS USED:

git log –oneline , git checkout -b feature1 notepad
lab1.txt git add lab1.txt git commit -m "Commit for
cherry-pick experiment" git checkout main git
cherry-pick feature1
git log --oneline



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git checkout master
Already on 'master'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git log --oneline source-branch
fatal: ambiguous argument 'source-branch': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git branch
  feature-branch
  main
* master

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git log --oneline feature-branch
17f065d (HEAD -> master, tag: v1.0, origin/main, origin/feature-branch, origin/HEAD, main, feature-
branch) Initial commit -adding a new file
cfbaade Initial commit

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$
```

EXPERIMENT 9: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

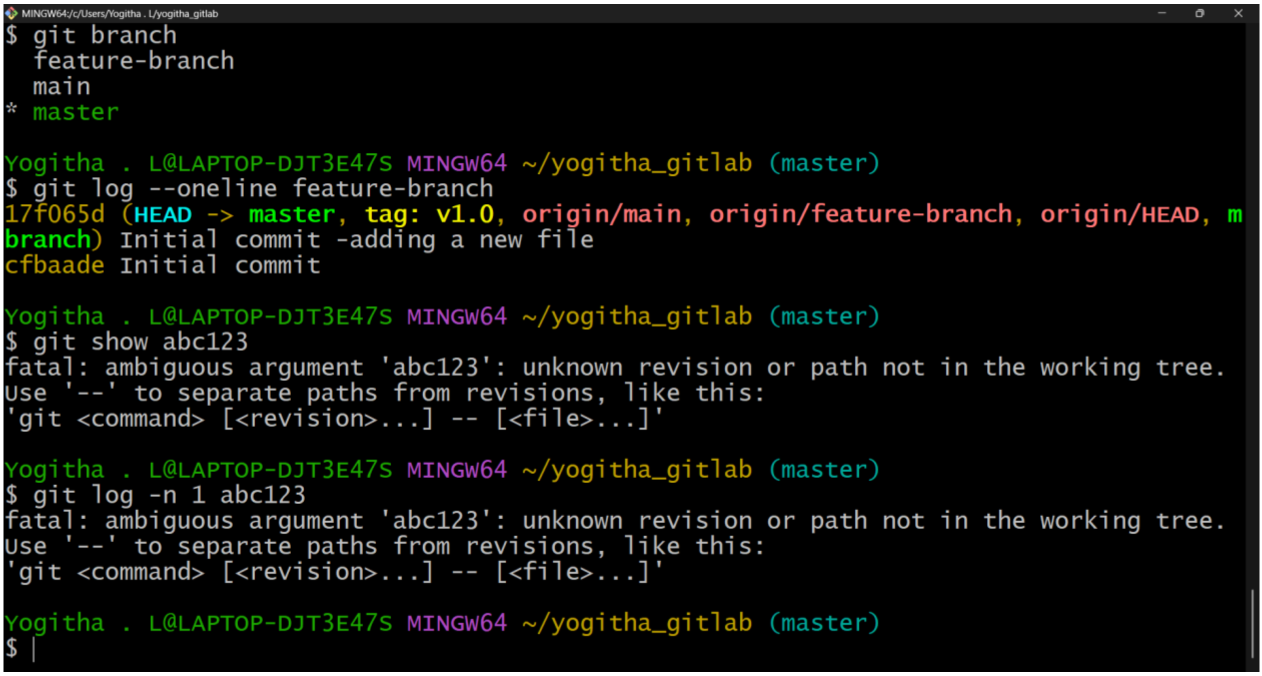
1. The commit history of the repository was viewed using the git log --oneline command to display concise commit information.
2. The complete commit history with author name, date, and commit message was viewed using the git log command.
3. A specific commit was selected from the history and its detailed information was displayed using the git show command.
4. The changes made in the selected commit were analyzed using the diff output.

COMMANDS USED:

git log –oneline , git

log ,

git show <commit-id>



```
$ git branch
  feature-branch
  main
* master

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git log --oneline feature-branch
17f065d (HEAD -> master, tag: v1.0, origin/main, origin/feature-branch, origin/HEAD, m
branch) Initial commit -adding a new file
cfbaade Initial commit

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git show abc123
fatal: ambiguous argument 'abc123': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git log -n 1 abc123
fatal: ambiguous argument 'abc123': unknown revision or path not in the working tree.
Use '--' to separate paths from revisions, like this:
'git <command> [<revision>...] -- [<file>...]'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ |
```

EXPERIMENT 10: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

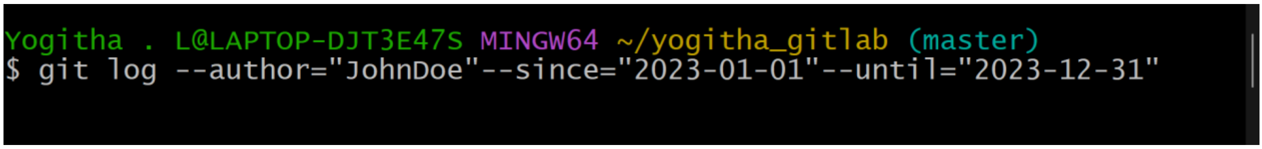
1. The current status of the repository was checked using the git status command.
2. The file lab1.txt was modified using Notepad.
3. The repository status was checked again to confirm that the file was modified.
4. The modified file was restored to its last committed state using the git restore lab1.txt command.
5. The repository status was verified to ensure that the working tree was clean.

COMMANDS USED:

git status

notepad lab1.txt

git status



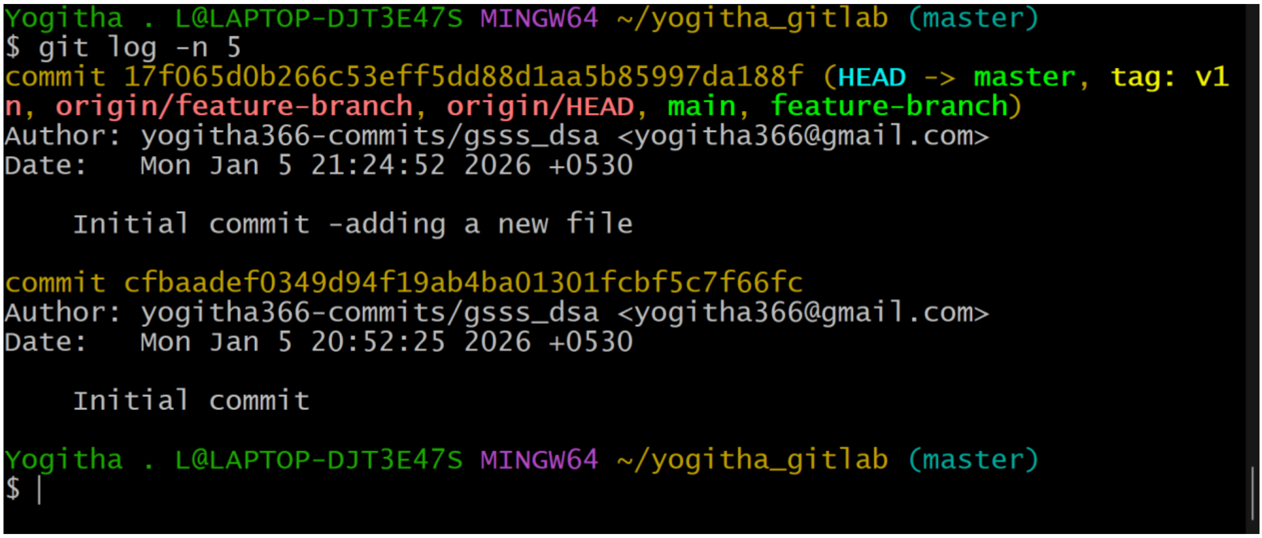
```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git log --author="JohnDoe"--since="2023-01-01"--until="2023-12-31"
```

EXPERIMENT 11: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

1. The current status of the local repository was checked using the git status command.
2. The committed changes were pushed to the remote GitHub repository using the git push origin main command.
3. After pushing, the status was verified to ensure the local and remote repositories were synchronized.
4. The updated files and commits were verified on the GitHub repository.

COMMAND USERS:
git status, git push origin



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git log -n 5
commit 17f065d0b266c53eff5dd88d1aa5b85997da188f (HEAD -> master, tag: v1
n, origin/feature-branch, origin/HEAD, main, feature-branch)
Author: yogitha366-commits/gsss_dsa <yogitha366@gmail.com>
Date:   Mon Jan 5 21:24:52 2026 +0530

    Initial commit -adding a new file

commit cfbaadef0349d94f19ab4ba01301fcbf5c7f66fc
Author: yogitha366-commits/gsss_dsa <yogitha366@gmail.com>
Date:   Mon Jan 5 20:52:25 2026 +0530

    Initial commit

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ |
```

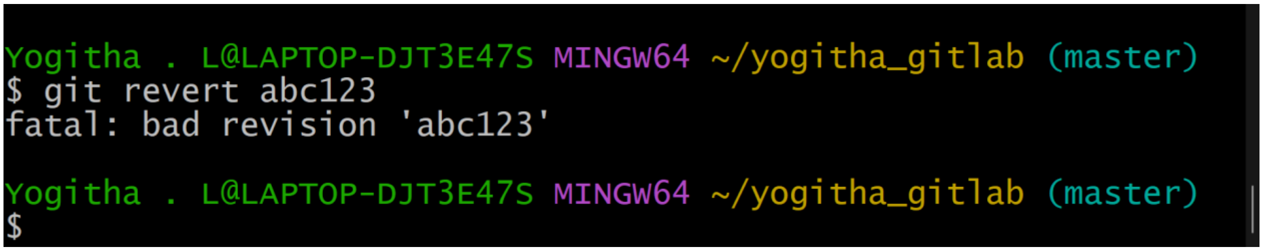
EXPERIMENT 12: ANALYSING AND CHANGING GIT HISTORY

PROCEDURE:

1. The commit history was viewed using the git log --oneline command.
2. The commit that needed to be undone was identified.
3. The git revert command was used to reverse the changes made by the selected commit.
4. A new revert commit was created without deleting previous commits.
5. The repository status and commit history were verified.
6. The revert commit was pushed to the remote GitHub repository

COMMANDS USED:

git log –oneline, git revert <commit-id>, git status, git pus git push origin main



```
Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$ git revert abc123
fatal: bad revision 'abc123'

Yogitha . L@LAPTOP-DJT3E47S MINGW64 ~/yogitha_gitlab (master)
$
```

git restore lab1.txt
git status